

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE  
BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES**

In re Application of: § Filed: November 24, 2003  
Richard D. Dettinger et al. §  
Serial No.: 10/720,963 § Group Art Unit: 2168  
Confirmation No.: 5212 § Examiner: Mahesh H. Dwivedi  
§  
For: METHOD FOR INVOKING AND INTERGRATING MULTIPLE FUNCTIONAL  
MODULES

MAIL STOP APPEAL BRIEF - PATENTS  
Commissioner for Patents  
P.O. Box 1450  
Alexandria, VA 22313-1450

**CERTIFICATE OF MAILING OR TRANSMISSION**

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Mail Stop Appeal Brief - Patents, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450, or facsimile transmitted to the U.S. Patent and Trademark Office to fax number 571-273-8300 to the attention of Examiner Mahesh H. Dwivedi, or electronically transmitted via EFS-Web, on the date shown below:

January 23, 2008  
Date

/Joseph Jong/  
Joseph Jong

**APPEAL BRIEF**

Dear Sir:

Applicants submit this Appeal Brief to the Board of Patent Appeals and Interferences on appeal from the decision of the Examiner of Group Art Unit 2168 dated August 22, 2007, finally rejecting claims 1-26. The final rejection of claims 1-26 is appealed. This Appeal Brief is believed to be timely since it is transmitted by the due date of January 23, 2008, as set by the filing of a Notice of Appeal on November 23, 2007.

Please charge the fee of \$510.00 for filing this brief to:  
Deposit Account No. 09-0465 / ROC920030278US1.

**TABLE OF CONTENTS**

1.	Identification Page.....	1
2.	Table of Contents .....	2
3.	Real Party in Interest .....	3
4.	Related Appeals and Interferences .....	4
5.	Status of Claims .....	5
6.	Status of Amendments .....	6
7.	Summary of Claimed Subject Matter .....	7
8.	Grounds of Rejection to be Reviewed on Appeal .....	10
9.	Arguments .....	11
10.	Conclusion .....	15
11.	Claims Appendix .....	16
12.	Evidence Appendix .....	21
13.	Related Proceedings Appendix .....	22

### **Real Party in Interest**

The present application has been assigned to International Business Machines Corporation, Armonk, New York.

### **Related Appeals and Interferences**

Applicant asserts that no other appeals or interferences are known to the Applicant, the Applicant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

### **Status of Claims**

Claims 1-26 are pending in the application. Claims 1-26 were originally presented in the application. Claims 1-26 stand finally rejected as discussed below. The final rejections of claims 1-26 are appealed. The pending claims are shown in the attached Claims Appendix.

### **Status of Amendments**

All claim amendments have been entered by the Examiner. No amendments to the claims were proposed after the final rejection.

## Summary of Claimed Subject Matter

Claimed embodiments include methods (*see claims* 1-14), computer programs stored on computer readable storage media (*see claims* 15-19) and computer systems (*see claims* 20-26) directed to processing a database query result that includes invoking a plurality of functional modules, each configured to process the query result. In particular, the claims are directed to a process where multiple functional modules may be invoked in a prescribed or derived sequence, without requiring data transformation. The prescribed or derived sequence may be contained in a configuration file accessible by the application. *See Application*, 1:11-14, Abstract.

### A. CLAIM 1 – INDEPENDENT

Claim 1 recites a method for invoking a plurality of functional modules configured to process a query result retrieved from a database from within an application. *See Application*, 1:11-14, 6:33-36, 7:1-7, 14:12-18. This method includes providing an interface for specifying the plurality of functional modules and providing a configuration file containing information regarding invocation of the functional modules. *See Application*, 7:9-15, 7:29-34, 8:1-4, Figure 1, configuration file 160, 10:22-28, 14:20-22, 16:33-36, 17:1-8. As claimed, the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules. *See Application*, 12:12-22, 17:18-25, Table 1, p. 15-16. This method also recites a step of receiving the query result retrieved from the database, *see Application*, 13:18-24, 14:21-22 Figure 3A, 302, Figure 2B result set 222, *et seq.*, and includes invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file. *See Application*, 12:12-22, Figure 2B, 161, 162, 13:18-24, Figures 3A and 3B, method step 304, sequence of plug-ins 310<sub>A</sub>, 16:1-11. This method also includes returning the processed query result to the application. *See Application*, 16:9-11.

### B. CLAIM 11 – INDEPENDENT

Claim 11 recites a method for invoking a plurality of specified functional modules configured to process a query result retrieved from a database from within an application. *See Application*, 1:11-14, 5:8-16, 6:28-31, 6:33-39 7:1-7, 14:12-18, Figure 3A, 3B, method step 302, 14:20-23, 16:16-18. This method includes a step of obtaining a set of one or more parameters required for invoking the specified functional modules, wherein at least one of the one or more parameters comprises a field of the query result. *See Application*, 14:27-32, 17:9-33, Figure 3B loop of method steps 306B-308, et seq., 18:1-2. This method also includes invoking one or more of the specified functional modules whose required parameters are available in a result set collection, wherein the result set collection is configured to store the query result and an output of each of the plurality of functional\_modules, as each of the plurality of functional modules is invoked to process the query result. *See Application*, 14:27-32, 17:9-33, Figure 3B loop of method steps 306B-308, et seq., 18:1-2 Figure 3A-3B, result set collection 309<sub>A-B</sub>. This method also includes obtaining a result set in response to invoking the one or more functional modules, adding the result set to the result set collection, and repeating steps (a) - (d) until all the specified functional modules have been executed. *See Application*, 14:27-32, 17:9-33, Figure 3B loop of method steps 306B-308, et seq., 18:1-2 Figure 3A-3B. This method also includes returning the processed query result to the application. *See Application*, 16:9-11.

#### C. CLAIM 15 – INDEPENDENT

Claim 15 is directed to a computer readable storage medium containing a program which, when executed, performs operations for invoking and integrating a plurality of functional modules configured to process a query result retrieved from a database from within an application. *See Application*, 1:11-14, 5:18-25, 8:9-20, 14:12-18, Figure 3A, 3B, method step 302, 14:20-23, 16:16-18. As claimed, the operation includes providing an interface for specifying the plurality of functional modules and providing a configuration file containing information regarding invocation of the functional modules. *See Application*, 7:9-15, 7:29-34, 8:1-4, Figure 1, configuration file 160, 10:22-28, 14:20-22, 16:33-36, 17:1-8. As claimed, the configuration file specifies at least an input field of the query result required by at least one of the functional



modules and at least one output field produced by one of the plurality of functional modules. *See Application*, 12:12-22, 17:18-25, Table 1, p. 15-16. The operation also includes receiving the query result retrieved from the database, *see Application*, 13:18-24, 14:21-22 Figure 3A, 302, Figure 2B result set 222, *et seq.*, and invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file. *See Application*, 12:12-22, Figure 2B, 161, 162, 13:18-24, Figures 3A and 3B, method step 304, sequence of plug-ins 310<sub>A</sub>, 16:1-11. This method also includes returning the processed query result to the application. *See Application*, 16:9-11.

#### D. CLAIM 20 – INDEPENDENT

Claim 20 is directed to a system having a processor and a plurality of functional modules, wherein each of the plurality of functional modules is configured to process a query result retrieved from a database. *See Application*, 5:27-33, 6:33-36, 7:1-7, 9:9-19, Figure 1, 160, *et seq.*, As claimed, the system also includes a configuration file containing information regarding execution of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules. *See Application*, 17:18-25. Figure 1, 161, 162, *et seq.*, 12:12-22, 17:18-25, Table 1, p. 15-16. As claimed, the system also includes an application from which the functional modules are accessible, wherein the application which, when executed by the processor, is configured to provide an interface for specifying a set of functional modules and execute the functional modules to process the query result in a manner determined according to information retrieved from the configuration file and present a user of the application with the processed query result. *See Application*, 1, 2A, 2B, application 120, 12:12-22, 13:14-24, 16:9-11, Figures 3A and 3B, method step 304, sequence of plug-ins 310<sub>A</sub>, 16:1-11.

### **Grounds of Rejection to be Reviewed on Appeal**

1. Rejection of claims 1-26 under 35 U.S.C. § 103(a) as being unpatentable over *Amro et al.*, U.S. Pat. No. 6,041,326 (hereinafter *Amro*) in view of *Young*, U.S. Pat. No. 6,560,606.

## **ARGUMENTS**

### ***Amro, in view of Young, Does Not Render Claims 1-26 Obvious under 35 U.S.C. § 103(a)***

#### *The Applicable Standard for Establishing a Case of Prima Facie Obviousness*

The Examiner bears the initial burden of establishing a *prima facie* case of obviousness. See MPEP § 2142. To establish a *prima facie* case of obviousness three basic criteria must be met. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one ordinary skill in the art to modify the reference or to combine the reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. See MPEP § 2143. The present rejection fails to establish at least the third criteria.

In this case, *Amro* does not teach a “method for invoking a plurality of functional modules configured to process a query result retrieved from a database from within an application” that includes “providing a configuration file containing information regarding invocation of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules,” and also includes “invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file,” as recited by claim 1. Claims 11, 15, and 20 recite similar limitations. Instead, *Amro* discloses a plug-in system configured to process a list of website links returned in response to a search engine query.

*Amro* disclose that a “search engine, on the Internet, is essentially a program that searches for keywords in files and documents found on the World Wide Web or other such computer networks.” See *Amro*, 9:14-16. *Amro* describes the use of a plug-in program configured to filter search results according to some criteria specified by the user. For example, *Amro* provides:

A user-defined plug-in program thus functions between the actual search engine utilized by the user and the user. Search results from the search engine can be filtered through such user-defined plug-in programs. The filtered search results are then displayed for the user as the actual search results.

*Amro*, 10:61-66. And further provides:

the user "plug-in" program acts as a filter by comparing the search engine "hits" with the database of known (i.e., previous) hits. Undesirable hits are thus weeded out in this manner, and the desirable hits (i.e., "good" hits) are presented, as illustrated at block 188. Finally, as illustrated at block 190, a "hit" list and ranking of such hits is presented to the user.

*Amro*, 11:63-67 – 12:1-2. The Examiner relies on these passages to assert that *Amro* discloses providing a configuration file containing information regarding invocation of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules, as recited by Claim 1. At best however, the passages cited by the Examiner reflect that a plug-in may be used to filter a list of websites returned by an internet search engine. By its own terms, *Amro* provides:

A simple comparison between a "hit" and this database may be programmed to weed out undesirable "hits." Also, the converse may also be programmed. More desirable network sites or types of networked sites may be ranked higher in a "hit" list produced than sites that are not as desirable to the university professor. For example, when searching for specific plant life indigenous to Zaire, sites actually residing the closest to Zaire can be ranked higher than sites located in Australia, for example.

*Amro*, 11:27-35. Nothing in these passages describe a configuration file that contains information specifying how the output of one functional module may be required as an input to another, and on this basis "invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file," as recited by the present claims. Instead, a filter is used to refine web-site results. Applicants submit, therefore, that the disclosure in *Amro* of a plug-in configured to refine a list of website links (through filtering and/or rank ordering) fails to disclose "a configuration file containing information regarding invocation of the functional modules," in particular, where "the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least

one output field produced by one of the plurality of functional modules.” Plainly, even assuming that the “filter” disclosed in *Amro* discloses a “configuration file,” in some abstract sense, such a “filter does not contain “information regarding invocation of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules.”

Further still, the Examiner concedes that *Amro* does not disclose “the plurality of functional modules,” as recited by claims 1, 11, 15, and 20. To address this limitation, the Examiner turns to *Young*. However, *Young* is directed to “telecommunication systems, and more particularly to computer processing of metered information.” *Young* 1:8-10. More simply, *Young* is directed to techniques for billing customers for using different telecommunication services. The material cited by the Examiner is directed to processing performed “pipeline stage” used to process an “input queue.” The pipeline itself is part of “a metered data processing system 104 for processing the metered data to generate useful information regarding communication services usage.” *Young*, 4:53-55. That is, the pipeline generates information used to calculate a customer bill. Depending on what services a consumer subscribes to, or consumes, the method for calculating a bill for a metered service may vary.

It is entirely unclear how the Examiner believes that the “pipeline stages” used to process an “input queue” to determine how much to charge a consumer for using a particular telecommunication service could be combined with a plug-in program configured to process a list of website links returned by a search engine. The Examiner suggests “Young’s [sic] would have allowed Amro’s [sic] to provide a method for improving efficiency in reducing overhead associated with processing, as noted by Young (Column 2, lines 32-35).” See Final Office Action, p. 10. Applicants submit, however, that this general goal of “improving efficiency in reducing overhead associated with processing,” fails to provide any specific indication of just how the proposed combination would operate, if it would operate at all. In an Advisory Action dated November 14, 2007, the Examiner repeats this same general conclusion:

However, *Young*’s method deals with multiple plug-ins with associated configuration files. The addition of *Young*’s method to *Amro*’s would allow

for multiple plug-ins to be accessed by a user to improve efficiency.  
*See Advisory Action*, continuation sheet. Respectfully, it is unclear as to what, exactly, is being improved in efficiency as suggested by the Examiner. *Amro* discloses that a user-defined filter may be used to filter a list of web-site hits returned by a search engine, and *Young* techniques for billing customers for using different telecommunication services. Other than the use of the word “plug-in” the techniques appear to be completely unrelated to one another.

Accordingly, for all the foregoing reasons, Applicants submit that claims 1, 11, 15, and 20, as well as the respective dependant claims, are believed to be allowable, and Applicants respectfully request that the Board vacate the final rejection and allow claims 1-26. .

### CONCLUSION

The Examiner errs in finding that claims 1-26 are unpatentable over *Amro* in view of *Young* under 35 U.S.C. § 103(a).

Withdrawal of the rejections and allowance of all claims is respectfully requested.

Respectfully submitted, and  
**S-signed pursuant to 37 CFR 1.4,**

/Gero G. McClellan, Reg. No. 44,227/

---

Gero G. McClellan  
Registration No. 44,227  
Patterson & Sheridan, L.L.P.  
3040 Post Oak Blvd. Suite 1500  
Houston, TX 77056  
Telephone: (713) 623-4844  
Facsimile: (713) 623-4846  
Attorney for Appellant(s)

## CLAIMS APPENDIX

1. (Previously Presented) A method for invoking a plurality of functional modules configured to process a query result retrieved from a database from within an application, comprising:
  - providing an interface for specifying the plurality of functional modules;
  - providing a configuration file containing information regarding invocation of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules;
  - receiving the query result retrieved from the database;
  - invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file; and
  - returning the processed query result to the application.
2. (Original) The method of claim 1, wherein the interface is a graphic user interface utilized by users to specify functional modules.
3. (Original) The method of claim 1, wherein the interface allows an external application to specify functional modules.
4. (Original) The method of claim 1, wherein:
  - the interface is utilized to specify a single multi-analysis functional module used to invoke the plurality of functional modules; and
  - the configuration file contains information relating the plurality of functional modules to the multi-analysis functional module.
5. (Original) The method of claim 1, wherein the configuration file contains an explicit sequence in which the plurality of functional modules should be executed.
6. (Original) The method of claim 1, wherein:



the configuration file contains information indicating one or more parameters required for invoking each of the functional modules; and

invoking the plurality of functional modules comprises invoking only those functional modules whose one or more required parameters are available.

7. (Original) The method of claim 1, wherein invoking the plurality of functional modules in a manner determined according to information retrieved from the configuration file comprises invoking at least two functional modules in parallel.

8. (Original) The method of claim 1, wherein the configuration file is in an extensible markup language (XML) format.

9. (Original) The method of claim 1, wherein at least one of the functional modules is a plug-in component of the application.

10. (Original) The method of claim 1, wherein at least one of the functional modules is an external application.

11. (Previously Presented) A method for invoking a plurality of specified functional modules configured to process a query result retrieved from a database from within an application comprising:

(a) obtaining a set of one or more parameters required for invoking the specified functional modules, wherein at least one of the one or more parameters comprises a field of the query result;

(b) invoking one or more of the specified functional modules whose required parameters are available in a result set collection, wherein the result set collection is configured to store the query result and an output of each of the plurality of functional modules, as each of the plurality of functional modules is invoked to process the query result;

(c) obtaining a result set in response to invoking the one or more functional modules;

- (d) adding the result set to the result set collection;
  - (e) repeating steps (a) - (d) until all the specified functional modules have been executed; and
  - (f) returning the processed query result to the application.
12. (Original) The method of claim 11, wherein the result set collection comprises results received in response to issuing a query.
13. (Original) The method of claim 11, wherein an interface is used to specify a plurality of functional modules by specifying a multi-analysis functional module.
14. (Original) The method of claim 13, wherein obtaining the set of one or more parameters required for invoking the specified functional modules comprises retrieving information from a configuration file relating the multi-analysis functional module to the specified functional modules.
15. (Previously Presented) A computer readable storage medium containing a program which, when executed, performs operations for invoking and integrating a plurality of functional modules configured to process a query result retrieved from a database from within an application, comprising:
- providing an interface for specifying the plurality of functional modules;
  - providing a configuration file containing information regarding invocation of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules;
  - receiving the query result retrieved from the database;
  - invoking the plurality of functional modules to process the query result in a manner determined according to information retrieved from the configuration file; and
  - returning the processed query result to the application.
16. (Previously Presented) The computer readable storage medium of claim 15,

wherein:

the interface is utilized to specify a single multi-analysis functional module used to invoke the plurality of functional modules; and

the configuration file contains information relating the plurality of functional modules to the multi-analysis functional module.

17. (Previously Presented) The computer readable storage medium of claim 15, wherein the configuration file contains an explicit sequence in which the plurality of functional modules should be executed.

18. (Previously Presented) The computer readable storage medium of claim 15, wherein:

the configuration file contains information indicating one or more parameters required for invoking each of the functional modules; and

invoking the plurality of functional modules comprises invoking only those functional modules whose one or more required parameters are available.

19. (Previously Presented) The computer readable storage medium of claim 15, wherein the configuration file is in an extensible markup language (XML) format.

20. (Previously Presented) A system, comprising:

a processor;

a plurality of functional modules, wherein each of the plurality of functional modules is configured to process a query result retrieved from a database;

a configuration file containing information regarding execution of the functional modules, wherein the configuration file specifies at least an input field of the query result required by at least one of the functional modules and at least one output field produced by one of the plurality of functional modules; and

an application from which the functional modules are accessible, wherein the application which, when executed by the processor, is configured to provide an interface for specifying a set of functional modules and execute the functional modules to process

the query result in a manner determined according to information retrieved from the configuration file and

present a user of the application with the processed query result.

21. (Original) The system of claim 20 wherein the application is a query building application.

22. (Original) The system of claim 20, wherein at least one of the plurality of functional modules is a plug-in component of the application.

23. (Original) The system of claim 20, wherein at least one of the plurality of functional modules is an external application.

24. (Original) The system of claim 20, wherein:  
the interface is utilized to specify a single multi-analysis functional module used to invoke the specified set of functional modules; and  
the configuration file contains information relating the specified set of functional modules to the multi-analysis functional module.

25. (Original) The system of claim 20, wherein the configuration file contains an explicit sequence in which the specified set of functional modules should be executed.

26. (Original) The system of claim 20, wherein:  
the configuration file contains information indicating one or more parameters required for invoking each of the specified set of functional modules; and  
invoking the specified set of functional modules comprises invoking only those functional modules whose one or more required parameters are available.

## EVIDENCE APPENDIX

None.

## RELATED PROCEEDINGS APPENDIX

None.